AMENDMENTS TO THE CLAIMS

1. (Currently Amended) A real-time OS simulator running on a general-purpose multi-thread OS, said simulator being operable to respectively that assigns assign a plurality of threads of the task processing thread to run on a general-purpose multi-thread OS to each of a plurality of tasks of to run on a real-time OS to be simulated, and simulates to simulate an operation of said the real-time OS on said the general-purpose multi-thread OS, said simulator comprising:

a system function, called by one of the plurality of threads of the general-purpose multi-thread OS which is assigned as a first task processing thread, operable to communicate with another of the plurality of threads of the general-purpose multi-thread OS for providing an instruction thereto so as to switch a state of the currently running task of the simulated real-time OS from that of the first task processing thread to that of a thread of the general-purpose multi-thread OS which is assigned as a second task processing thread; and

task switching instruction means for receiving a request issued from said task processing thread under same conditions as in said real-time OS, and providing an instruction for switching the tasks in response to said request; and

a task switching thread for operable to receive from said system function the instruction to switch a state of the currently running task of the simulated real-time OS by making selected one of said task processing threads run by suspending and resuming a plurality of said task processing threads of the general-purpose multi-thread OS corresponding to task processing threads with according to the capabilities of said the general-purpose multi-thread OS in cooperation with said task switching instruction means.

2. (Currently Amended) The real-time OS simulator according to claim 1, wherein said system function task switching instruction means selects is operable to select a task processing thread to run next, provides to provide the instruction to said task switching thread for switching the currently running first task processing thread to the thread of the general-purpose multi-thread OS which is assigned as the selected task

- 4 -

processing thread~~tasks to said task switching thread~~, and to then ~~suspends~~ suspend the first task processing thread ~~that has issued said request~~, and

~~in response to the instruction,~~ said task switching thread ~~resumes~~ is operable to, in response to the instruction provided by said system function, resume the selected task processing thread after the first ~~a preceding running~~ task processing thread is suspended.

3.   (Currently Amended) The real-time OS simulator according to claim 2, wherein ~~in response to the instruction for switching the tasks,~~ said task switching thread, in response to the instruction for switching the tasks, is operable to check, ~~checks~~ at predetermined intervals, whether the first task processing thread ~~preceding running task thread~~ is suspended or not.

4.   (Currently Amended) The real-time OS simulator according to claim 1, wherein said system function ~~task switching instruction means selects~~ is operable to select a task processing thread to run next, ~~provides~~ to provide the instruction to said task switching thread for switching the currently running task to the selected task processing thread~~tasks to said task switching thread~~, and to then ~~sets~~ set the thread of the general-purpose multi-thread OS corresponding to the first task processing thread that has issued ~~said~~ the request in a waiting state, and

~~in response to the instruction,~~ said task switching thread ~~suspends~~ is operable to, in response to the instruction, suspend the thread of the general-purpose multi-thread OS corresponding to the first task~~a preceding running task processing thread~~, and then ~~releases~~ to release the selected task processing thread from a previously set ~~the~~ waiting state for resuming.

5.   (Currently Amended) The real-time OS simulator according to claim 2, wherein said system function ~~task switching instruction means provides~~ is operable to provide the instruction to said task switching thread after said task switching thread has been indicated as being ~~is~~ enabled to start processing.

6.   (Currently Amended) The real-time OS simulator according to claim 1, wherein

said system function ~~task switching instruction means provides~~ is operable to provide the instruction to said task switching thread after having selected the second ~~selecting a~~ task processing thread as the task processing thread to run next, and

said task switching thread ~~runs~~ is operable to run with a higher priority than said task processing threads and, in response to the instruction, ~~suspends~~ to suspend a preceding running task processing thread and then ~~resumes~~ resume the selected second task processing thread.

7.   (Original) The real-time OS simulator according to claim 1, further comprising task processing thread creating means for creating said task processing thread.

8.   (Currently Amended) The real-time OS simulator according to claim 1, wherein
        a further one of the plurality of threads of the general-purpose multi-thread OS is assigned as an exception handling thread ~~for corresponding to~~ task exception handling of each of ~~said~~ the tasks ~~and running on said multi-thread OS is further assigned to each of said tasks~~, and

said task switching thread ~~selects~~ is operable to select a thread to run next from among said task processing threads and said exception handling threads.

9.   (Currently Amended) The real-time OS simulator according to claim 8, further comprising thread creating means for creating said task processing threads and said exception handling thread.

10.   (Currently Amended) The real-time OS simulator according to claim 1, further comprising interrupt handling means for receiving an interrupt request issued by an interrupt thread that generates a pseudo-interrupt, suspending a running task processing thread, calling an interrupt handler corresponding to the received interrupt request, and then selecting a task processing thread to run next for resuming.

11.   (Currently Amended) The real-time OS simulator according to claim 10, wherein when said interrupt handling means receives ~~receiving~~ the interrupt request from ~~said~~ the

interrupt thread while another interrupt thread is running, said interrupt handling means suspends the running interrupt thread, calls the interrupt handler corresponding to the interrupt request, and then resumes the suspended interrupt thread.

12. (Currently Amended) The real-time OS simulator according to claim 10, wherein said the interrupt thread includes a system clock interrupt thread that generates a pseudo-interrupt at predetermined time intervals.

13. (Currently Amended) The real-time OS simulator according to claim 10, further comprising interrupt thread creating means for creating said the interrupt thread.

14. (Currently Amended) A computer-readable recording medium for recording a program to run on a computer, the program being operable to perform for a simulation method of assigning threads of a general-purpose multi-thread OS to a task processing threads of thread to run on a general-purpose multi-thread OS to each of a plurality of tasks to run on a real-time OS to be simulated, and simulating an operation of said real-time OS on said general-purpose multi-thread OS, said simulation method comprising the steps of:

receiving a request issued from a thread of the general-purpose multi-thread OS which is assigned as a said task processing thread; under same conditions as said real-time OS and

providing an instruction to another thread of the general-purpose multi-thread OS which is assigned as task switching thread for switching the tasks in response to said the request received in said receiving of the request; and

making a selected thread of the general-purpose multi-thread OS which is assigned as a one of said task processing threads thread run by suspending and resuming said the selected task processing thread according to the task processing threads with capabilities of said the general-purpose multi-thread OS.

15. (Currently Amended) The recording medium according to claim 14, wherein

in said ~~instruction~~ providing ~~step~~ of the instruction, a task processing thread to run next is selected, and then the task processing thread that had ~~has~~ issued ~~said~~ the request is suspended, and

in said making of the selected task processing thread run ~~step~~, the selected task processing thread is resumed after the task processing thread that had issued the request ~~a preceding running thread~~ is suspended.

16. (Currently Amended) The recording medium according to claim 14, wherein
in said ~~instruction~~ providing ~~step~~ of the instruction, a task processing thread to run next is selected, and then the task processing thread that had ~~has~~ issued ~~said~~ the request is set to a waiting state, and

in said making of the selected task processing thread run ~~step~~, after the task processing thread that issued the request ~~a preceding running task processing thread~~ is suspended, a previously set waiting state of the selected task processing thread is cleared for resuming.

17. (Currently Amended) The recording medium according to claim 14, wherein
in said ~~instruction~~ providing ~~step~~ of the instruction, a task processing thread to run next is selected, and

~~said run step~~ suspending the task processing thread issuing the request and resuming the selected task processing thread is given a higher priority than running ~~said~~ the task processing threads, and, in said making of the selected task processing thread run ~~step~~, after the task processing thread issuing the request ~~a preceding running task processing thread~~ is suspended, the selected task processing thread is resumed.

18. (Currently Amended) The recording medium according to claim 14, wherein
a thread of the general-purpose multi-thread OS is assigned as an exception handling thread for ~~corresponding to task~~ exception handling of each of ~~said~~ the tasks ~~and running on said multi-thread OS is further assigned to each of said tasks~~, and

in said making of the selected task processing thread run step, a thread to run next is selected from among said the task processing threads and said the exception handling thread.

19.   (Currently Amended) The recording medium according to claim 14, wherein said simulation method further comprises the step of receiving an interrupt request issued from an interrupt thread that generates a pseudo-interrupt, suspending a running task processing thread, calling an interrupt handler corresponding to the received interrupt request, and then selecting a task processing thread to run next for resuming.

20.   (Currently Amended) A program for being operable to perform a simulation method of assigning threads of a general-purpose multi-thread OS to a task processing threads of thread to run on a general-purpose multi-thread OS to each of a plurality of task to run on a real-time OS to be simulated, and simulating an operation of said the real-time OS on said the general-purpose multi-thread OS, said simulation method comprising the steps of:

    receiving a request issued from a thread of the general-purpose multi-thread OS which is assigned as a said task processing thread; under same conditions as said real-time OS and

    providing an instruction to another thread of the general-purpose multi-thread OS which is assigned as task switching thread for switching the tasks in response to said the request received in said receiving of the request; and

    making a selected thread of the general-purpose multi-thread OS which is assigned as a one of said task processing threads thread run by suspending and resuming said the selected task processing thread according to the task processing threads with capabilities of said the general-purpose multi-thread OS.

21.   (Currently Amended) The program according to claim 20, wherein
    in said instruction providing step of the instruction, a task processing thread to run next is selected, and then the task processing thread that had has issued said the request is suspended, and

in said <u>making of the selected task processing thread</u> run ~~step~~, the selected task processing thread is resumed after <u>the task processing thread that had issued the request</u> ~~a preceding running thread~~ is suspended.

22.   (Currently Amended) The program according to claim 20, wherein

in said ~~instruction~~ providing ~~step~~<u>of the instruction</u>, a task processing thread to run next is selected, and then the task processing thread that <u>had</u> ~~has~~ issued ~~said~~ <u>the</u> request is set to a waiting state, and

in said <u>making of the selected task processing thread</u> run ~~step~~, after <u>the task processing thread that issued the request</u> ~~a preceding running task processing thread~~ is suspended, a <u>previously set</u> waiting state of the selected task processing thread is cleared for resuming.

23.   (Currently Amended) The program according to claim 20, wherein

in said ~~instruction~~ providing ~~step~~ <u>of the instruction</u>, a task processing thread to run next is selected, and

~~said run step~~ <u>suspending the task processing thread issuing the request and resuming the selected task processing thread</u> is given a higher priority than running ~~said~~ <u>the</u> task processing threads, and, in said <u>making of the selected task processing thread</u> run ~~step~~, after <u>the task processing thread issuing the request</u> ~~a preceding running task processing thread~~ is suspended, the selected task processing thread is resumed.

24.   (Currently Amended) The program according to claim 20, wherein

<u>a thread of the general-purpose multi-thread OS is assigned as</u> an exception handling thread <u>for</u> ~~corresponding to task~~ exception handling of each of ~~said~~ <u>the</u> tasks ~~and running on said multi-thread OS is further assigned to each of said tasks~~, and

in said <u>making of the selected task processing thread</u> run ~~step~~, a thread to run next is selected from among ~~said~~ <u>the</u> task processing threads and ~~said~~ <u>the</u> exception handling thread.

25.   (Currently Amended) The program according to claim 20, wherein said simulation method further comprises ~~the step of~~ receiving an interrupt request issued from an interrupt thread that generates a pseudo-interrupt, suspending a running task processing thread, calling an interrupt handler corresponding to the <u>received</u> interrupt request, and then selecting a task processing thread to run next for resuming.

26.   (Currently Amended) The real-time OS simulator according to claim 4, wherein said <u>system function</u> ~~task switching instruction means provides~~ <u>is operable to provide</u> the instruction to said task switching thread after said task switching thread <u>has been indicated as being</u> ~~is~~ enabled to start processing.

27.   (New) The real-time OS simulator according to claim 8, further comprising a thread creating unit operable to create said task processing threads and said exception handling thread.

28.   (New) The real-time OS simulator according to claim 1, further comprising an interrupt handling unit operable to receive an interrupt request issued by an interrupt thread that generates a pseudo-interrupt, to suspend a running task processing thread, to call an interrupt handler corresponding to the received interrupt request, and to then select a task processing thread to run next for resuming.

29.   (New) The real-time OS simulator according to claim 10, wherein when said interrupt handling unit receives the interrupt request from the interrupt thread while another interrupt thread is running, said interrupt handling unit is operable to suspend the running interrupt thread, to call the interrupt handler corresponding to the received interrupt request, and to then resume the suspended interrupt thread.

30.   (New) The real-time OS simulator according to claim 10, wherein the interrupt thread includes a system clock interrupt thread that generates a pseudo-interrupt at predetermined time intervals.

31.   (New) The real-time OS simulator according to claim 10, further comprising an interrupt thread creating unit operable to create the interrupt thread.